

Speed-up of heuristic search and hypothetical reasoning

メタデータ	言語: jpn 出版者: 公開日: 2017-10-05 キーワード (Ja): キーワード (En): 作成者: メールアドレス: 所属:
URL	http://hdl.handle.net/2297/16673

氏名	越野 亮
生年月日	
本籍	石川県
学位の種類	博士(工学)
学位記番号	博甲第664号
学位授与の日付	平成16年3月25日
学位授与の要件	課程博士(学位規則第4条第1項)
学位授与の題目	ヒューリスティック探索と仮説推論の高速化
論文審査委員(主査)	木村 春彦(工学部・教授)
論文審査委員(副査)	船田 哲男(工学部・教授) 伊藤 俊次(工学部・教授) 山根 智(工学部・教授) 南保 英孝(工学部・講師)

学位論文要旨

This thesis proposes two new bi-directional approximate heuristic search methods and two new speed-up methods of the inconsistent process and the subsumption process on hypothetical reasoning deal with predicate logic and a new approximate search method and a new optimal search on cost-based hypothetical reasoning. Heuristic search is a universal problem-solving mechanism in Artificial Intelligence (AI). The aim of heuristic search is to find a solution (path) from an initial state to a goal state. Real-Time A* (RTA*) is a representative heuristic search algorithm for finding approximate solutions. The Multi-State Commitment (MSC) method was introduced into RTA* and dramatically improved the performance in problems such as the N-puzzle. As well, Real-Time Bidirectional Search (RTBS) also improved RTA* by changing a unidirectional search into a bidirectional one. First proposed method is that introduces MSC into RTBS. Second proposed method is Bidirectional Depth-First A* search (BDFA*) using backtrack technique, which are efficient finding solutions methods. Hypothetical reasoning is a knowledge process framework, which can find explanations for a given set of observation by assuming some hypothetical sets. KICK-HOPE achieved efficient reasoning, however, the inconsistent process and the subsumption process still consume much calculation time. Third and fourth proposed method improved that vain processes could reduce. Cost-based hypothetical reasoning is to find a best explanation for a given observation based on the assumption of a set of hypothesis. Previously efficient method that utilized the A* algorithm was presented, however, the time and space computational complexity are exponential. Fifth proposed method is that Real-Time A* (RTA*) was introduced to cost-based hypothetical reasoning. Sixth proposed method is to change Depth-First A* search.

1 序論

本論文では人工知能の研究の一分野である問題解決の代表的な手法である「ヒューリスティック探索」と「仮説推論」において、筆者がこれまでにやってきた高速化手法について述べる。

2 ヒューリスティック探索の高速化

第2章では、状態空間グラフにおけるヒューリスティック探索手法として2つの両方向近似解探索アルゴリズムを提案する。ヒューリスティック探索とは、問題に関するヒューリスティックな知識を評価関数として用いることで効率的に初期状態から目標状態までの経路(解)を求めることである。推論など人工知能の最も基本となる問題であるが、計算複雑度はNP-完全またはNP-困難であり、状態空間のサイズが大きくなると現実的な時間で解を求めることが困難になる。そこで、時間をかけて最適解を求めることよりも、許容可能な時間内に近似解を求めることが重要となる。そのような近似解探索アルゴリズムとして、RTA* (実時間探索: Real-Time A*)がある。RTA*は、先読み探索と移動を交互に行うアルゴリズムであり、先読みにより探索範囲を限定することで、計算量は改善される。しかし、次に展開する状態を一つに限定するため、選択を誤った場合の悪影響は大きい。多状態コミットメント探索(MSC: Multi-State Commitment)では、RTA*による探索でn個の状態を保持することで、そのような問題点を解消している。一方、初期状態と目標状態から同時に両方向から実時間探索を実行する実時間両方向探索(RTBS: Real-Time Bidirectional Search)では、初期状態と

目標状態から出発した2つの問題解決器が、常に他方の問題解決器を目指して移動することで効率化を実現している。

2.1 多状態実時間両方向探索

実時間両方向探索に多状態コミットメントを導入した多状態実時間両方向探索 (MSRTBS : Multi-State Real-Time Bidirectional Search) を提案する。

前向き探索で生成された状態を保持するリストを $OPEN_1$, 計算用のリストを $TEMP_1$ とする。同様に、後向き探索で生成された状態を保持するリストを $OPEN_2$, 計算用のリストを $TEMP_2$ とする。以下にアルゴリズムを示す。

Step1: 初期状態 s_0 を $OPEN_1$, 目標状態 G を $OPEN_2$ に加える。保持する状態の数 n を設定する。

Step2: if ($OPEN_1$ =空 または $OPEN_2$ =空) then 探索失敗, 終了。

Step3: $OPEN_1$ の先頭の状態を x , $OPEN_2$ の先頭の状態を y とする。

Step4: if ($x = y$) then 探索成功, 終了。

Step5: if ($TEMP_1$ =空) then

1. $OPEN_1$ から先頭の状態 x を取り出す。
2. x を展開し生成される全ての子状態の集合と $OPEN_1$ に含まれる状態の集合を $TEMP_1$ に加える。
3. すべての $x' \in TEMP_1$ に関して $h(x', y)$ を計算する。

Step6: if ($TEMP_2$ =空) then

1. $OPEN_2$ から先頭の状態 y を取り出す。
2. y を展開し生成される全ての子状態の集合と $OPEN_2$ に含まれる状態の集合を $TEMP_2$ に加える。
3. すべての $y' \in TEMP_2$ に関して $h(x, y')$ を計算する。

Step7: 以下の式に従い, $h(x, y)$ を更新する。

$$h(x, y) \leftarrow \text{secondmin}_{x', y'} \begin{cases} c(x, x') + h(x', y) \\ c(y, y') + h(x, y') \end{cases}$$

($h(x, y)$ の値を $c(x, x') + h(x', y)$ と $c(y, y') + h(x, y')$ のうち2番目に小さい値に更新する。)

Step8: 前向き探索と後向き探索の制御戦略

1. if ($\min\{h(x', y)\} < \min\{h(x, y')\}$) then
 - (a) 全ての x' を $OPEN_1$ とし, 昇順にソートし, $TEMP_1$ を空にする。(最小のものが複数個あれば最小のものをランダムに並び替える。)
 - (b) if ($OPEN_1$ の個数 $> n$) then n 個になるまで, $OPEN_1$ の中で最大の評価値の状態を削除する。
2. if ($\min\{h(x', y)\} > \min\{h(x, y')\}$) then
 - (a) 全ての y' を $OPEN_2$ とし, 昇順にソートし, $TEMP_2$ を空にする。(最小のものが複数個あれば最小のものをランダムに並び替える。)
 - (b) if ($OPEN_2$ の個数 $> n$) then n 個になるまで, $OPEN_2$ の中で最大の評価値の状態を削除する。
3. if ($\min\{h(x', y)\} = \min\{h(x, y')\}$) then ランダムで (1)(2) を実行する。

Step9: Step2 へ

63 パズルにおける, 提案手法 MSRTBS と従来の多状態コミットメント探索 MSC-RTA* と両方向実時間探索 RTBS の実行時間 (秒), 解の長さ, 実行ステップ数, 記憶節点数の平均値を表 1 に示す。なお, 実行ステップ数とは問題解決器が状態を移動した回数であり, 記憶節点数とは探索した問題空間の広さである。なお, 63 パズルは 8×8 の盤面に 1 から 63 までの番号のついたタイルと一つの空白があり, タイルをランダムな状態から, 番号が正しく並び目標状態へとタイルをスライドさせてゆく問題である。

表 1: 63 パズルにおける実験結果

手法	実行時間 (秒)	解の長さ	ステップ数	記憶節点数
MSC-RTA*	2422	9107	176071	78055
RTBS	1713	10883	117278	88271
MSRTBS	118	6564	43668	22162

実験結果より, 本提案手法である MSRTBS は, 従来の MSC-RTA* や RTBS と比べ, 大幅に速度・解の精度が向上していることがわかる。これは, 実行ステップ数と記憶節点数が少なくなったことから, 探索空間が狭められたためであると考えられる。

2.2 両方向深さ優先 A*探索

提案手法 MSRTBS は N パズルにおいて、従来の MSC や RTBS と比べ、大幅に性能を改善できたが、迷路問題においては有効ではないという結果が得られた。これは、実時間探索の性質として、障害物が多い迷路問題では、一度判断を誤ると問題解決者は何度も学習と移動を繰り返さないと目標に至ることができないという問題があるためである。そこで、バックトラックを用いることで、何度も学習と移動を繰り返さなくても高速に解を見つけることができる両方向深さ優先 A*探索 (BDFA*: Bidirectional Depth-First A*) を提案する。

前向き探索で生成された状態を保持するリストを $OPEN_1$ 、展開した状態を保持するリストを $CLOSED_1$ 、後向き探索で生成された状態を保持するリストを $OPEN_2$ 、展開した状態を保持するリストを $CLOSED_2$ とする。一時的に展開した状態を保持するリストを $TEMP_1, TEMP_2$ とし、以下に BDFA* のアルゴリズムを示す。なお、終了条件は一方の探索でもう一方の探索空間にたどり着いたら探索成功とする。

Step1: 初期状態 s_0 を $OPEN_1$ 、目標状態 G を $OPEN_2$ に加える。

Step2: if ($OPEN_1 = \text{空}$ または $OPEN_2 = \text{空}$) then 探索失敗, 終了。

Step3: $OPEN_1$ の先頭の状態 x とする。 $OPEN_2$ の先頭の状態 y とする。

Step4: if ($x \in CLOSED_2$ または $y \in CLOSED_1$) then 探索成功, 終了。

Step5: x を展開し、子ノードの集合を x' とする。 y を展開し、子ノードの集合を y' とする。

Step6: すべての x' に関して $f(x', y) = c(s_0, x') + h(x', y) + c(y, G)$ を計算、昇順にソートし (もし、同じ値が複数個あればランダムでソート順を変える)、 $x' \in CLOSED_1$ でなければ $TEMP_1$ に入れる。

すべての y' に関して $f(x, y') = c(s_0, x) + h(x, y') + g(y', G)$ を計算、昇順にソートし (もし、同じ値が複数個あればランダムでソート順を変える)、 $y' \in CLOSED_2$ でなければ $TEMP_2$ に入れる。

Step7: 前向き探索と後向き探索の制御戦略

(1) if ($\min(f(x', y)) < \min(f(x, y'))$) then $OPEN_1$ の先頭を取り出し $CLOSED_1$ に入れる。 $TEMP_1$ を $OPEN_1$ に加える。

(2) if ($\min(f(x', y)) > \min(f(x, y'))$) then $OPEN_2$ の先頭を取り出し $CLOSED_2$ に入れる。 $TEMP_2$ を $OPEN_2$ に加える。

(3) if ($\min(f(x', y)) = \min(f(x, y'))$) then ランダムで (1)(2) を実行する。

Step8: $TEMP_1, TEMP_2$ を空にする。

Step9: Step2 へ。

格子状グラフにおいて、迷路のサイズを 400×400 として通過することのできない障害物を 35% の割合でランダムに配置した問題を 100 個用意し、両方向深さ優先 A*探索 (BDFA*), 単方向深さ優先 A*探索 (DFA*), 実時間両方向探索 (RTBS), 実時間探索 (RTA*) 実行時間 (msec), 解の長さ, 実行ステップ数, 探索空間の大きさの平均を表 2.2 に示す。

表 2: 迷路 (400×400) における実験結果

	BDFA*	DFA*	RTBS	RTA*
実行時間	8	61	4417	1174
解の長さ	1570	1668	1069	1058
ステップ数	2218	23718	922144	353783
探索空間の大きさ	2318	14542	1740	1681

BDFA* は実行時間において、RTA* や RTBS と比べ大幅な速度向上を実現していることがわかる。これは、同じ状態の訪問を削減することにより、実行ステップ数が減ったためであると考えられる。解の質においては、RTA* や RTBS と比べ低下している。これは、BDFA* では、バックトラックが生じると、他方の探索の目標状態が離れた場所になるため、探索空間が広がり経路が長くなったと考えられる。

3 仮説推論の高速化

第 3 章では、述語論理を扱う仮説推論の矛盾処理と包摂処理の高速化手法を提案する。仮説推論とは、真か偽か不明な事柄をとりあえず真と考えて (仮説を立てて) 推論を進め、矛盾なく観測事象 (ゴール) が説明できれば、立てた仮説は正しかったと考える形式の推論である。従来の仮説推論システムである KICK-HOPE は、後向き探索と前向き探索を組み合わせて効率的な推論を実現している。しかし、推論途中で支持している仮説集合に矛盾が生じていないかを調べる矛盾処理と冗長な仮説集合が生じていないかを調べる包摂処理に膨大な計算時間がかかることが問題となっている。KICK-HOPE では、仮説のビットベクトル表現とビット演算処理により高速化を実現しているが、無駄な処理を行っていることがわかった。そこで、矛盾処理・包摂処理が生じる条件から、仮説合成の際に、簡単な矛盾処理・包摂処理が必要かどうかチェックを行い、無駄な矛盾処理・包摂処理を削減する手法を提案する (提案手法 3)。簡単な例題知識ベー

ス、回路の故障診断、時間割作成のスケジューリング問題などにおいて比較実験を行い、大幅に速度向上を実現したことを示す。また、事例ベースとよばれる学習手法を導入した高速化手法を提案する（提案手法4）。

4 コストに基づく仮説推論の高速化

第4章では、コストに基づく仮説推論の高速化手法を提案する。仮説推論では一般に、説明を与える仮説集合（解）は複数存在する。しかしながら、診断、設計問題などにおいては、ユーザが要求する解はすべての説明ではなく、むしろ最も好ましい説明であることが多い。各仮説にコスト（確率や信頼度から求める）を付け最も好ましい説明（最小コストの説明）を求める推論をコストに基づく仮説推論という。

4.1 実時間探索を用いた仮説推論システム

コストに基づく仮説推論は最適解探索問題に置き換えることができ、A*アルゴリズムを適用することができる。A*アルゴリズムは最適解を求める探索アルゴリズムとしては、効率が最も良いが、実用規模の問題では、膨大なメモリと時間が必要になる。そこで、探索アルゴリズムの近似解法として用いられている実時間探索を導入したシステムを提案する。以下にアルゴリズムを示す。

- Step1（初期設定）知識ベース KB とゴール g を与える。先読みの深さ l を設定する。 KB に含まれる矛盾知識を取り除き、述語の引数を無視した $\overline{KB} \setminus \overline{IC}$ に抽象化する。矛盾知識 inc の最小の仮説集合を求める。現在の探索ノード x を g とし、 SG をノードのリストとする。
- Step2（展開） x と KB における一段階のSLD導出で得られるサブゴールを全て求めて SG に追加する。
- Step3（終了判定） $SG = \phi$ （サブゴールがないとき） x の仮説集合を解として出力し、終了する。
- Step4（矛盾処理） SG に現れる仮説が矛盾知識の仮説集合を含むかどうかを調べ、矛盾するサブゴールを SG から削除する。
- Step5（先読み探索）最小の評価値 $\min\{f(x) = g(x) + \hat{h}(x) | x \in SG\}$ を持つノードを x とする。
- Step6（先読み深さチェック）もし、ノード x の深さが先読み深さ l なら SG を削除し Step2 へ。それ以外なら、 SG を保持し Step2 へ。

4.2 深さ優先 A*探索を用いた仮説推論システム

A*アルゴリズムを用いた仮説推論システムは、命題論理に抽象化した知識をヒューリスティックな知識として用いることで、効率的に探索を実現したが、A*アルゴリズムは最適解の可能性のある未探索ノードをすべて保存する必要があるため、メモリ量が膨大になるという問題が生じる。それに伴い、膨大になった未探索ノードのリストから最小コストのノードを検索する時間が膨大になるという問題がある。

まず、A*アルゴリズムを深さ優先 A* に改良し、高速に準最適解（最適解の保証のない解）を探し、次に、その解を暫定解として、後戻り（バックトラック）を起こし、分枝限定法の要領で最適解の可能性のない暫定解より評価値が悪いノードを枝刈りしながら、最適解探索に必要なノードがなくなるまで、繰り返し探索することで、最適解を求めることができる。以下にアルゴリズムを示す。

- Step1（初期設定）知識ベース KB とゴール g を与える。現在の探索ノード x を g とし、 SG , SG_{tmp} をノードのリストとする。暫定解 $solution$ のコストを ∞ とする。制限時間 T を設定する。
- Step2（終了条件 1）制限時間 T を超えた場合、現在の暫定解 $solution$ を出力し、終了する。
- Step3（展開） x と KB における一段階のSLD導出で得られるサブゴールを全て求めて、その集合を SG_{tmp} とする。
- Step4（矛盾処理） SG_{tmp} に現れる仮説が矛盾知識の仮説集合を含むかどうかを調べ、矛盾するサブゴールを SG_{tmp} から削除する。
- Step5（枝切り）暫定解 $solution$ のコストより SG_{tmp} の持つノードのコストが大きい場合は、最適解に達する可能性のないノードのため削除する。
- Step6（暫定解の更新） $SG_{tmp} = \phi$ のとき、 x の仮説集合を暫定解 $solution$ とする。最大の深さ $\max(\text{depth}(SG))$ を持つノードの集合を SG_{tmp} とする。ただし、 $\text{depth}()$ はノードの深さを返す関数である。
- Step7（終了条件 2） $SG = \phi$ （最適解探索に必要なノードがない）場合、 x の仮説集合を最適解 $solution$ として出力し、終了する。
- Step8（選択）最小の評価値 $\min\{f(x) = g(x) + \hat{h}(x) | x \in SG_{tmp}\}$ を持つノードを x とし、 SG_{tmp} から x を取り除き、 SG_{tmp} を SG に追加する。Step2 に戻る。

5 結論

本論文では、問題解決の方法として最も基礎的な状態空間と記号論理を対象として、様々な探索手法と仮説推論の高速化手法を提案した。

ヒューリスティック探索の近似解法として、多状態実時間探索 (MSRTBS) と両方向深さ優先 A*探索 (BDFA*) を提案した。MSRTBS は N パズルにおいて大幅な速度向上と精度向上を実現したが、迷路問題においては速度、解の質ともに悪くなるという結果が得られた。BDFA* は迷路問題においては大幅な速度向上を実現したが、N パズルでは速度、解の質ともに悪くなるという結果が得られた。

述語論理を扱う仮説推論において、全探索の KICK-HOPE を対象に、矛盾処理、包摂処理の高速化、事例ベース推論を導入した高速化手法を提案した。また、仮説に好ましさの基準 (コスト) を付けたコストに基づく仮説推論において、実時間探索 (RTA*) を用いた手法、深さ優先 A*探索を用いた手法を提案した。

学位論文審査結果の要旨

平成 16 年 1 月 27 日に第 1 回学位論文審査委員会を開催、1 月 28 日に口頭発表、その後第 2 回審査委員会を開催し、慎重審議の結果以下の通り判定した。なお、口頭発表における質疑を最終試験に代えるものとした。

本論文は、大きく分けるとヒューリスティック探索の高速化と仮説推論の高速化に分けられる。前者は、問題に関するヒューリスティック (経験に培われた) な知識を評価関数として用いることで効率的に初期状態から目標状態までの経路 (解) を求めることであり、最適解の計算複雑度は NP 完全または NP 困難である。そのため、状態空間のサイズが大きくなると現実的な時間で解を求めることが困難となり、許容可能な時間内で近似解を求めることが要求される。申請者はこのような状況の下で、多状態実時間両方向探索と両方向深さ優先 A*探索という近似解を求める手法を提案した。後者では、真か偽か不明な事柄をとりあえず真と考えて (仮説を立てて) 推論を進め、矛盾なく観測事象が説明できれば、立てた仮説が正しかったと考える仮説推論を対象に、KICK-HOPE の矛盾処理と包摂処理の高速化、事例ベース推論の導入による高速化、実時間探索を用いた仮説推論システム、深さ優先 A*探索を用いた仮説推論システムを提案した。

以上の研究成果は、探索と仮説推論の高速化に大きく貢献するものであり、本論文は博士 (工学) に値するものと判定した。